Mastery UML

Stephen Espinal

Controllers

HomePageController

DisplayHomePage @GetMapping

BlogPostsController
+ addBlogPost @Postmapping
+ DisplayPage @GetMapping
+ EditBlogPost @PostMapping
+ DeleteBlogPost @PostMapping
+ Add HashTag (from blog content property using substring , contains StringUtils)

AdminController
+ addCategory @PostMapping
+ addUser @PostMapping
+ displayAdminPage(): Model
+ edit User @PostMapping

CategoryController
+ addCategory @PostMapping
+ displayCategories @GetMapping
+ deleteCategory @PostMapping
+ edit User @PostMapping

blogPost
+ int blogId
+ String blogTitle
+ String blogContent
+ String isPublished
+ int userId

HashTag
+ int hashTagId
+ String hashTagName

Category
+ int categoryId
+ String categoryName

User
+ int id
+ String username
+ String password
+ Boolean enabled

Role
+ int id
+ String roleName

Services

BlogPostsService
+ createBlogPost(BlogPost post): BlogPost
+ getBlogPostById(int postId): BlogPost
+ getAllBlogPosts(): List<BlogPost>
+ updateBlogPost(BlogPost post): BlogPost
+ deleteBlogPost(int postId): BlogPost

HashTagsService
+ createHashTag(HashTag hashTag): HashTag
+ getHashTagById(int hashTagId): HashTag
+ getAllHashTags(): List<HashTag>
+ updateHashTag(HashTag hashTag): HashTag
+ deleteHashTag(int hashTagId): HashTag

CategoryService
+ createRole(Category category): Category
+ getCategoryById(int categoryId): Category
+ getAllCategory(): List<Category>
+ updateCategory(Category category): Category
+ deleteCategory(int categoryId): Category

UserService
+ createUser(User user): User
+ getUserById(int userId): User
+ getAllUser(): List<User>
+ updateUser(User user): User
+ deleteUser(int userId): User

RoleService
+ createRole(Role role): Role
+ getRoleById(int roleId): Role
+ getAllRole(): List<Role>
+ updateRole(Role role): Role
+ deleteRole(int roleId): Role

DTO's
+ BlogPosts
+ HashTags
+ Category
+ Users
+ Roles

Data Access Objects

BlogPostsDao
- final JdbcTemplate jdbcTemplate
+ createBlogPost(BlogPost post): BlogPost
+ getBlogPostById(int postId): BlogPost
+ getAllBlogPosts(): List<BlogPost>
+ updateBlogPost(BlogPost post): BlogPost
+ deleteBlogPost(int postId): BlogPost

HashTagsDao
- final JdbcTemplate jdbcTemplate
+ createHashTag(HashTag hashTag): HashTag
+ getHashTagById(int hashTagId): HashTag
+ getAllHashTags(): List<HashTag>
+ updateHashTag(HashTag hashTag): HashTag
+ deleteHashTag(int hashTagId): HashTag

CategoryDao
- final JdbcTemplate jdbcTemplate
+ createRole(Category category): Category
+ getCategoryById(int categoryId): Category
+ getAllCategory(): List<Category>
+ updateCategory(Category category): Category
+ deleteCategory(int categoryId): Category

UserDao
- final JdbcTemplate jdbcTemplate
+ createUser(User user): User
+ getUserById(int userId): User
+ getAllUser(): List<User>
+ updateUser(User user): User
+ deleteUser(int userId): User

RoleDao
- final JdbcTemplate jdbcTemplate
+ createRole(Role role): Role
+ getRoleById(int roleId): Role
+ getAllRole(): List<Role>
+ updateRole(Role role): Role
+ deleteRole(int roleId): Role

Data Persistence: MySQL